CSC249/449 HW3: Tutorial for Discriminant Correlation Filter Based Visual Tracking

Jing Shi
University of Rochester
Rochester, 14627, USA
j.shi@rochester.edu

1. Introduction

1.1. Problem Statement

Given a target, estimate the target state over time. In this tutorial, the target is the location of an patch in the first frame of a video, the output is the position of the target in the following frames.

1.2. Discriminant Correlation Filters

Discriminant Correlation Filters (DCF) enjoy the advantages of both high speed and high accuracy. Since MOSSE [3] proposed the first DCF for visual tracking, which runs at high speed of more than 600 fps, many other subsequent DCF-related tracking algorithms occurs. We are particularly interested on how the DCF can be integrated into deep neural network while keeping high speed, so we will systematically study one of the DCF variants called DCFNET [7]. In the following content, we will mainly learn

1. How DCF can be used for tracking.
2. How DCF tracker is updated online.

In order to understand the above content, some extra knowledge we need to master is

1. Discrete Fourier Transformation and its properties.
2. Parseval's theorem.
3. Matrix trace and its derivative.
4. Derivative of a real valued function w.r.t. complex variable.

Although seems daunting, you will fall in love with the clever design of the tracker after the digestion of the content. Furthermore, you are required to implement such tracker after the theoretic part.

2. Methods

2.1. Overview of DCF for tracking

Here is how the DCF tracker works. A deep neural network $\phi(\cdot)$ is used to extract the feature of image. In current frame, we have the feature of target patch $\phi(\mathbf{x}) \in \mathbb{R}^{M\times N\times D}$, and the idea response $\mathbf{y} \in \mathbb{R}^{M\times N}$ which is a gaussian function peaked at the center. We expect to learn an optimal filter $\mathbf{w} \in \mathbb{R}^{M\times N\times D}$ that can output $\mathbf{y}$ given $\phi(\mathbf{x})$. Then we crop a search patch z in the new frame, which is a patch centered at the same position with $\mathbf{x}$, and obtain the features $\phi(\mathbf{z})$. Next, we apply the filter $\mathbf{w}$, which is a robust representation of the target patch in the previous frame, to do cross correlation on the search patch $\phi(\mathbf{z})$ to generate the correlation response map $\mathbf{g}$. Finally the target translation can be estimated by searching the maximum value of correlation response map $\mathbf{g}$.

The desired filter $\mathbf{w}$ can be obtained by minimizing the output ridge loss:

$$\epsilon = ||\sum_{l=1}^{D} \phi_l(\mathbf{x}) \star \mathbf{w}_l - \mathbf{y}||^2 + \lambda \sum_{l=1}^{D} ||\mathbf{w}_l||^2, \tag{1}$$

where $\mathbf{w}^l$ refers to the channel l of the filter $\mathbf{w}$, $\star$ means circular correlation (explained in App. A), the constant $\lambda \geq 0$ is regularization coefficient and $||\cdot||$ represent Frobenius norm (explained in App. B).

2.2. Optimization of DCF

It is easy to optimize such $\mathbf{w}$ using gradient descent. However, it is feasible to get the analytical optimal solution for $\mathbf{w}$ in spectrum domain. Using the relationship of convolution and correlation (App. A), parseval's theorem (see App. C) , convolution theorem and time reverse property (see App. D), we can rewrite Eq. (1) as

$$\epsilon = \frac{1}{MN}\big(||\sum_{l=1}^{D} \Phi_l(\mathbf{x}) \odot W_l^* - Y||^2 + \lambda \sum_{l=1}^{D} ||W_l||^2\big) \tag{2}$$

where the upper case variable $\Phi_l(\mathbf{x}), W_l, Y_l$ represent the Fourier transform of their lower case counterparts, and $(\cdot)^*$ means conjugate, $\odot$ denotes Hadamard product. The optimization is conducted by setting its derivative w.r.t. W^* to be zero following the same steps in Sec. 2.4. The optimization process is left as an assignment in Sec. 3.1. Finally, the solution can be gained as

$$W_l = \frac{\Phi_l(\mathbf{x}) \odot Y^*}{\sum_{k=1}^{D} \Phi_k(\mathbf{x}) \odot \Phi_k^*(\mathbf{x}) + \lambda}, \tag{3}$$

where $\frac{(\cdot)}{(\cdot)}$ is element-wise division. Hence $\mathbf{w}_l$ can be obtained as $\mathcal{F}^{-1}(W_l)$, where $\mathcal{F}^{-1}(\cdot)$ means inverse Fourier transform.

2.3. Inference with DCF

The inference process will fix the parameter in $\Phi(\cdot)$, but will update the DCF W. The specific steps are described as follows

Update (when frame $t = 1$):

1. Initialize the target patch at the first frame, enlarge the target patch and crop it as $\mathbf{x}^{(1)}$.
2. Update the accumulated $\hat{\Phi}(\mathbf{x}^{(1)}) = \Phi(\mathbf{x}^{(1)})$.
3. Calculate the optimal DCF as $W^{(1)}$ by using Eq. (3), but replace the $\Phi_l(\mathbf{x}^{(1)})$ with $\hat{\Phi}_l(\mathbf{x}^{(1)})$.
4. Update the accumulated DCF as $\hat{W}^{(1)} = W^{(1)}$

Update (when frame $t \neq 1$):

1. Given the target patch $\mathbf{x}^{(t)}$ at the frame $t, (t \neq 1)$.
2. Update the accumulated $\hat{\Phi}(\mathbf{x}^{(t)}) = (1 - lr) \times \hat{\Phi}(\mathbf{x}^{(t-1)}) + lr \times \Phi(\mathbf{x}^{(t)})$.
3. Calculate the optimal DCF as $W^{(t)}$ by using Eq. (3), but replace the $\Phi_l(\mathbf{x}^{(t)})$ with $\hat{\Phi}_l(\mathbf{x}^{(t)})$.
4. Update the accumulated DCF as $\hat{W}^{(t)} = (1 - lr) \times \hat{W}^{(t-1)} + lr \times W^{(t)}$

Forward:

1. For $\forall t$, crop the box in frame $t + 1$ as $\mathbf{z}^{(t+1)}$ at the same position as $\mathbf{x}^{(t)}$ in frame t.
2. Get the response $\mathbf{g}^{(t+1)}$ by using

$$\mathbf{g}^{(t+1)} = \mathcal{F}^{-1}\big(\sum_{l=1}^{D} \hat{W}_l^{(t)*} \odot \Phi_l(\mathbf{z}^{(t+1)})\big). \tag{4}$$

3. Finally the target's translation from $\mathbf{x}^{(t)}$ to $\mathbf{x}^{(t+1)}$ can be estimated by searching the maximum value of correlation response map $\mathbf{g}^{(t+1)}$, thus $x^{(t+1)}$ is obtained.

For each frame, started from $t = 1$, alternate **update** and **forward**.
So, when the DCF-based tracker is in test phase, it can update the DCF every frame, so it can do on-line updating.

2.4. MOSSE filters

Minimum Output Sum of Squared Error (MOSSE) filter. Practice of optimization for a real valued function of a complex variable. Objective function is

$$\min_{W^*} \sum_i ||F_i \odot W^* - Y_i||^2, \tag{5}$$

where matrix F_i, Y_i represent i th input feature the Gaussian output, respectively; matrix W^* represents the complex conjugate of the learnable filter; $\odot$ denotes Hadamard product and $||\cdot||$ indicates Frobenius norm (explained in App. B). The optimization steps are described as follows

1. Rewrite the objective function using trace

$$\begin{aligned} &\sum_i ||F_i \odot W^* - Y_i||^2 \\ =&\sum_i tr[(F_i \odot W^* - Y_i)^H (F_i \odot W^* - Y_i)] \\ =&\sum_i [tr((F_i^H \odot W^T F_i \odot W^*) - tr(F_i^H \odot W^T Y_i) - tr(Y_i^H F_i \odot W^*) + tr(Y_i^H Y_i)], \end{aligned} \tag{6}$$

   where $(\cdot)^H$ indicates conjugate transpose, and for convenience, we allow Hadmard product to have higher priority then matrix product.

2. According to [6], we can optimize Eq. (6) by setting its partial w.r.t W^* equal to zero while treating W as an independent variable. It is good for you to know why W and W^* can be treated as independent variable by refering the short tutorial on this technique in [6], because you might think $\frac{dz^*}{dz}$ is ill-defined if taken MTH 282.

$$\begin{aligned} 0 &= \frac{\partial}{\partial W^*} \sum_i [tr((F_i^H \odot W^T F_i \odot W^*) - tr(F_i^H \odot W^T Y_i) - tr(Y_i^H F_i \odot W^*) + tr(Y_i^H Y_i)] \\ &= \sum_i (F_i \odot F_i^* \odot W - F_i \odot Y_i^*). \end{aligned} \tag{7}$$

   If you have problem deducing Eq. (7), please refer [4] to know the derivative of trace. We can finally distribute the summation and solve for W as

$$W = \frac{\sum_i F_i \odot Y_i^*}{\sum_i F_i \odot F_i^*}. \tag{8}$$

3. Assignment

3.1. Optimization of DCF

Prove Eq. (3) is the optimal solution for Eq. (2).
Hint: mimic the steps in Sec. 2.4.

3.2. Proving Parseval's theorem for 2-d DFT

Prove the Parseval's theorem for 2-d DFT:

$$\sum_{m=0}^{M-1} \sum_{n=1}^{N-1} |x[m,n]|^2 = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} |X[m,n]|^2. \tag{9}$$

Hint: firtly prove Eq. (24).

References

[1] Wikipedia kronecker product. https://en.wikipedia.org/wiki/Kronecker_product. Accessed: 2019-01-28. 5

[2] Wikipedia vectorization (mathematics). https://en.wikipedia.org/wiki/Vectorization_(mathematics). Accessed: 2019-01-28. 5

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 1
[4] W. Fischer and K. Meier-Hellstern. The markov-modulated poisson process (mmpp) cookbook. *Performance evaluation*, 18(2):149–171, 1993. 3
[5] R. M. Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006. 7
[6] D. Messerschmitt et al. Stationary points of a real-valued function of a complex variable. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-93*, 2006. 3
[7] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017. 1

Appendices

A. Connection between Circular Convolution and Cross Correlation

Let $*$ denote circular convolution and $\star$ indicate cross correlation, the relationship between $*$ and $\star$ of two 1-d signal $x[n]$ and $y[n]$ is

$$x[n] * y[n] = x[n] \star y[-n]. \tag{10}$$

Hence the circular convolution form of Eq. (1) is

$$\epsilon = || \sum_{l=1}^{D} \phi_l(\mathbf{x})[m,n] * \mathbf{w}_l[-m,-n] - \mathbf{y}||^2 + \lambda \sum_{l=1}^{D} ||\mathbf{w}_l||^2. \tag{11}$$

Also, by default, if the index of 1-d signals exceeds $[0, N-1]$, then $x[n] = x[(n \bmod N)]$. The similar rule holds for 2-d signal.

B. Frobenius Norm

The Frobenius norm $||\cdot||_F$ is a matrix norm of an $M \times N$ matrix A defined as the square root of the sum of the absolute squares of its elements

$$||A||_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} |A_{ij}|^2}, \tag{12}$$

where $|\cdot|$ means modules of a complex number. It is also equal to the square root of the matrix trace of AA^H, where A^H is the conjugate transpose, i.e.

$$||A||_F = tr(AA^H). \tag{13}$$

C. Parseval's Theorem

Theorem C.1. *For 1-d finite discrete signals x and X with length N having the relation $X = Fx$, where $F \in \mathbb{C}^{N\times N}$ is an orthogonal-like transformation matrix satisfying $F^H F = NI$, the following equation holds*

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2. \tag{14}$$

Proof.

$$\sum_{k=0}^{N-1} |X[k]|^2 = X^H X = x^H F^H F x = N x^H x = N \sum_{n=0}^{N-1} |x[n]|^2. \tag{15}$$

□

For Fourier transformation Eq. (17), the transform matrix D_N satisfying $D_N^H D_N = NI$ (see App. D.1), thus Parseval' theorem holds for Fourier transformation.

D. Discrete Fourier Transformation (DFT)

D.1. Definition of 1-d DFT

For a 1-d finite discrete signal $x[n]$ with length N, its Discrete Fourier Transformation $X[k]$ is defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jnk\frac{2\pi}{N}}. \tag{16}$$

For simplicity, we define $W_N = e^{-j\frac{2\pi}{N}}$, so that we can rewrite Eq. (16) as

$$X = D_N x, \tag{17}$$

where

$$D_N = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & W_N^1 & \cdots & W_N^{N-2} & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & W_N^{N-2} & \cdots & W_N^{(N-2)^2} & W_N^{(N-2)(N-1)} \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-2)} & W_N^{(N-1)^2} \end{pmatrix}. \tag{18}$$

We need to notice that D_N satisfies $D_N^H D_N = NI$, here is the proof

Proof. Let $Q = D_N^H D_N$, so we have

$$\begin{aligned} Q[k1, k2] &= \sum_{n=0}^{N-1} D_N^*[k_1, n] D_N[k_2, n] \\ &= \sum_{n=0}^{N-1} W_N^{-k_1 n} W_N^{k_2 n} \\ &= \sum_{n=0}^{N-1} W_N^{(k2-k1)n} = \sum_{n=0}^{N-1} e^{-j\frac{2\pi(k_2-k_1)n}{N}}, \end{aligned} \tag{19}$$

hence we get

$$Q[k1, k2] = \begin{cases} N & k1 = k2, \\ 0 & k1 \neq k2. \end{cases} \tag{20}$$

So, $Q = NI$ □

Therefore, the inverse DFT is easily obtained by the inverse of Eq. (17)

$$x = \mathcal{F}^{-1}(X) = D_N^{-1} X = \frac{1}{N} D_N^H X. \tag{21}$$

D.2. Definition of 2-d DFT

For a 2-d finite discrete signal $x[m, n]$ with shape $M \times N$, its DFT is defined as

$$X[k, l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] e^{-j2\pi(\frac{km}{M} + \frac{ln}{N})}. \tag{22}$$

In order to represent 2-d DFT in matrix form, we use the vectorization operation 'vec' defined in [2] to vectorize 2-d signal into 1-d, and Eq. (22) can be rewritten as

$$\text{vec}(X) = (D_N \otimes D_M)\text{vec}(x), \tag{23}$$

where $\otimes$ is Kronecker products [1] and D_N follows the definition in Eq. (18). Also, $D_N \otimes D_M$ share the similar orthogonal property as D_N:

$$(D_N \otimes D_M)^H (D_N \otimes D_M) = MNI. \tag{24}$$

You are required to prove Eq. (24) in Sec. 3.2.

The inverse DFT is deduced as

$$\text{vec}(x) = \mathcal{F}^{-1}(\text{vec}(X)) = (D_N \otimes D_M)^{-1}\text{vec}(X) = \frac{1}{MN}(D_N^H \otimes D_M^H)\text{vec}(X). \tag{25}$$

D.3. Convolution Theorem

Here we only talk the circulate convolution, and show the case in 1-d situation, which is easily extensible to 2-d situation. Given two 1-d signals $x[n]$ and $y[n]$ with length N, and their DFT $X[k]$ and $Y[k]$, the convolution theorem is stated as

$$\mathcal{F}(x * y) = X \odot Y. \tag{26}$$

Proof. We firstly rewrite $x * y$ in the matrix form as

$$x * y = \begin{pmatrix} y[0] & y[N-1] & y[N-2] & \cdots & y[1] \\ y[1] & y[0] & y[N-2] & \cdots & y[2] \\ y[2] & y[1] & y[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y[N-1] & y[N-2] & y[N-3] & \cdots & y[0] \end{pmatrix} \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{pmatrix} = C_y x. \tag{27}$$

According to App. E, $C_y^T = U\Lambda_y U^H$, where $\Lambda_y = \text{diag}(Y)$, and $\text{diag}(\cdot)$ can turn a vector to diagonal matrix. Notice that $D_N^T = D_N$, we have

$$C_y = (U\Lambda_y U^H)^T = \frac{1}{N} D_N^* \Lambda_y D_N^T = \frac{1}{N} D_N^H \Lambda_y D_N. \tag{28}$$

So

$$\mathcal{F}(x * y) = D_N(C_y x) = D_N(\frac{1}{N} D_N^H \Lambda_y D_N x) = \Lambda_y X = X \odot Y. \tag{29}$$

□

D.4. Time Reverse

Time reverse property is stated as

$$\mathcal{F}(x[-n]) = X^*[k] \tag{30}$$

Proof. According to App. E, build circular matrix C_x from x as

$$C_x = \begin{pmatrix} x[0] & x[1] & \cdots & x[N-1] \\ x[N-1] & x[0] & \cdots & x[N-2] \\ \vdots & \vdots & \ddots & \vdots \\ x[1] & x[2] & \cdots & x[0] \end{pmatrix} = U\Lambda_x U^H. \tag{31}$$

Also, the time-reversed x is expressed as $\hat{x} = [x[0], x[N-1], \cdots, x[1]]^T$. Build circular matrix $C_{\hat{x}}$ from $\hat{x}$ as

$$C_{\hat{x}} = \begin{pmatrix} x[0] & x[N-1] & \cdots & x[1] \\ x[1] & x[0] & \cdots & x[2] \\ \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[N-2] & \cdots & x[0] \end{pmatrix} = U\Lambda_{\hat{x}} U^H. \tag{32}$$

Hence we have $\Lambda_x = U^H C_x U$ and $\Lambda_{\hat{x}} = U^H C_x^T U$. Noticing x is a real-valued signal so that $C_x^H = C_{\hat{x}}$, we obtain

$$\Lambda_x^* = \Lambda_x^H = U^H C_x^H U = U^H C_{\hat{x}} U = \Lambda_{\hat{x}}. \tag{33}$$

Hence $X^* = \hat{X}$

□

E. Circulate Matrix

Given a 1-d signal $c = [c_0, c_1, \cdots, c_{N-1}]^T$ the circulate matrix C derived from c is written as

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-2} & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-3} & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_2 & c_3 & \cdots & c_0 & c_1 \\ c_1 & c_2 & \cdots & c_{N-1} & c_0 \end{pmatrix}. \tag{34}$$

An important property of C is that it can be diagonalized as

$$C = U \Lambda U^H, \tag{35}$$

where $U = \frac{1}{\sqrt{N}} D_N$, and Λ is the diagonal matrix whose diagonal elements are the DFT of c. The proof will not be covered in this tutorial but can be found in Sec. 3.1 in [5].